

## TP final : Réalisation d'un Chat/Blog avec commentaires

L'objectif de ce TP est de mélanger les principales notions vues en cours : interactions avec une base de données et modèle CRUD, gestion des sessions et contrôle d'accès ainsi que l'architecture MVC.

2 niveaux de difficultés sont proposés. Le premier niveau ne permet pas d'obtenir une note supérieure à 13.

### Niveau 1 (projet « allégé ») :

Vous devez réaliser un mini chat.

Le projet comportera donc 2 tables : utilisateurs, messages.

La page d'accueil permet de visualiser les 10 derniers messages de la discussion. Il n'y a qu'une seule discussion (qui correspond à la liste séquentielle des messages postés)

- Il existe 2 types d'utilisateurs : les non logués et les logués
- Les utilisateurs non logués peuvent voir la discussion
- Les utilisateurs logués peuvent voir la discussion et poster un nouveau message
- Chaque message possède une date de post et un contenu.
- Lorsqu'un message est posté, son auteur est nécessairement l'utilisateur logué
- Chaque message doit afficher le nom de l'utilisateur qui a posté le message ainsi que la date du post
- Un utilisateur doit pouvoir s'inscrire, se connecter et se déconnecter, ajouter une photo de profil
- Un bouton doit permettre de masquer / visualiser tous les messages

En plus de l'aspect fonctionnel de votre projet vous serez notamment évalué sur :

- La clarté du code
- La sécurité (mot de passe haché, requêtes préparées)
- L'intuitivité du parcours utilisateur et l'ergonomie générale

### Niveau 2 (projet « standard ») :

Vous devez réaliser un mini blog avec la possibilité, pour les utilisateurs logués, de laisser des commentaires. Le projet comportera donc 3 tables : utilisateurs, billets et commentaires.

Il existe 2 types d'utilisateurs : les non logués et les logués. De plus, le "propriétaire" a des droits spécifiques dans la mesure où c'est le seul à pouvoir publier un nouveau billet. Un utilisateur doit pouvoir s'inscrire, se connecter et se déconnecter, ajouter une photo de profil.

Les principales contraintes sont :

- Un utilisateur est identifié par un login/mot de passe.

- Seul le propriétaire peut poster un nouveau billet (on vérifiera que les données stockées en session correspondent bien à celles du propriétaire).

- Par défaut les commentaires sont cachés. Sur chaque billet un lien/bouton permet de visualiser uniquement le billet cliqué ainsi que l'ensemble des commentaires associés.

- Un utilisateur non logué peut uniquement consulter les billets et afficher/masquer tous les commentaires associés à un billet.

idée de réalisation : On ramènera en une seule requête les billets et les commentaires associés et on cachera dans un div les commentaires. On pourra également ne ramener que les billets et ramener les commentaires en ajax si l'utilisateur clique sur un lien ou un bouton pour les afficher.

- Chaque billet possède une date de post et un contenu. Son auteur est nécessairement le propriétaire du blog.

- Chaque commentaire est également identifié par un auteur et une date de post.

- Seul un utilisateur logué peut ajouter un commentaire à un billet. Chaque commentaire se réfère donc à un billet et comporte une information sur la date de post, l'auteur et un contenu.

- Par défaut sur la page de garde du site les 3 derniers billets apparaissent. L'ensemble des billets est également consultable dans une section "archives".

Interface d'administration :

- Le propriétaire du site doit pouvoir modifier/supprimer/des billets

- le propriétaire du site peut également gérer les utilisateurs et les commentaires (modifier/supprimer).

Vous développerez un maximum de fonctionnalités dans le temps imparti.

En plus de l'aspect fonctionnel de votre projet vous serez notamment évalué sur :

- La clarté du code
- La sécurité (mot de passe haché)
- Le respect de l'architecture MVC
- L'intuitivité du parcours utilisateur et l'ergonomie générale

**Pour les 2 niveaux ! livrables :**

- l'url de l'application, hébergée et fonctionnelle (avec un jeu de données d'exemple, le login/mdp d'une utilisation test ainsi que le login/mdp du propriétaire du blog pour la version standard)

MMI 2024-2025  
Renaud Eppstein  
R3.13

- un zip avec les codes sources de l'application et un dump de la base de données à déposer sur e-learning dans la zone de dépôt dédiée

-une notice indiquant notamment

- le login/mdp du propriétaire (pour la version 2)

- d'éventuelles notes et commentaires précisant ce qui est développé partiellement ou ne fonctionne pas

Le rendu devra être déposé sur elearning dans un espace de dépôt dédié. Merci à ceux ayant des disques cryptés (Mac) de déposer un zip directement exploitable. Un malus sanctionnera les notices incomplètes (le mot de passe du propriétaire n'est pas fourni, l'url d'accès au back office n'est pas indiquée etc...)